

Orientierung am Konkreten

Expertensysteme mit Gedächtnis

Thomas J. Schult

Würden Sie sich mit einem schwierigen Problem jemandem anvertrauen, der sein Gedächtnis verloren hat? Kaum vorstellbar – doch erstellten Software-Entwickler zur Lösung komplexer Problemstellungen bisher Expertensysteme, die kein Gedächtnis besitzen. Das soll jetzt anders werden.

Der Mann hat einen skeptischen Gesichtsausdruck, ein Praktiker mit langjähriger Erfahrung und hochgekrempelten Ärmeln. 'Give him 10 years of experience overnight?', fragt er ungläubig angesichts eines kleinen PS/2 neben ihm. 'No problem', antwortet der smarte Verkäufer an seiner Seite, vermutlich ein Außendienstmitarbeiter der Firma Inference. Denn die Szene findet sich in der amerikanischen Fachpresse auf ganzseitigen Anzeigen eben dieses Softwarehauses, das als Hersteller von Expertensystemshells bekannt wurde.

Ebenso ungläubig schaut sicher auch der informierte Leser, weiß er doch, daß der Aufwand für die Erstellung eines Expertensystems in der Regel in Mannjahren zu messen ist. Doch das ist Vergangenheit, jedenfalls nach Ansicht von Inference und anderen Softwareschmieden, die innerhalb weniger Monate nachzogen. Expertensysteme kann man auch ganz anders machen, lautet ihre Botschaft: lernfähig, flexibler, schneller, menschenähnlicher. 'Fallbasiert' ist das neue Zauberwort. Fälle sind der zentrale Bestandteil dieser neuen Technologie zur Wissensverarbeitung. Gemeint sind konkrete Beispiele für expertenhaftes Verhalten, die im System gespeichert und bei einer ähnlichen Situation 'erinnert' und genutzt werden.

Regel oder Fall?

Zeichneten sich etwa medizinische Expertensysteme bisher durch eine große Fülle sehr spezieller Wenn-Dann-Regeln aus, die aus ihrem Entstehungszusammenhang abstrahiert wurden, so sieht ein fallbasiertes System in diesem Bereich ganz anders aus: Ein Fall enthält nur Informationen über einen ganz bestimmten Patienten, mit persönlichen Daten, Krankheitssymptomen, Therapien und ihrer Wirkung. Das System hat die Aufgabe, große Mengen solcher Fälle bereitzuhalten und bei Eingabe von neuen Patientendaten und -symptomen einen Therapievorschlag aufgrund seiner Kenntnis von ähnlichen Krankheitsfällen zu machen, um die ärztliche Arbeit zu unterstützen. Die Leistungsfähigkeit eines solchen Systems steht und fällt natürlich mit der Qualität des Gedächtnisses, das heißt mit der Anzahl der gespeicherten Fälle und ihrer Indizierung aufgrund eines Ähnlichkeitsmaßes.

Bis ins Detail

Warum erscheint es sinnvoll, bei Expertensystemen noch einmal ganz von vorne anzufangen und die grundlegende Architektur zu revidieren? Die Hauptaufgabe bei der traditionellen Expertensystementwicklung besteht in der Identifizierung von Regeln. Zentraler Bestandteil von Expertensystemshells ist daher immer ein Inferenz-

mechanismus gewesen, der die Anwendung und Verkettung von Regeln zur Problemlösung durchführt – ein komplexer Suchvorgang. Steht das System später einmal vor der gleichen Aufgabenstellung, ist der gleiche Aufwand wiederum erforderlich, denn Lernen ist nicht vorgesehen.

Aber das größte Problem liegt wohl darin, die erforderlichen Regeln erst einmal zu finden, sie in einer Programmiersprache auszudrücken und sie später auch bei geänderten Anforderungen geeignet anzupassen – Fragen der Wissensakquisition und der Wissensmodifikation also. Ersteres wird sogar allgemein als der entscheidende Grund dafür angesehen, daß der Expertensystemtechnologie der Durchbruch auf breiter Front bisher versagt blieb. Die Tools waren zwar da, aber der Aufwand für das 'knowledge engineering', die Übertragung von Expertenwissen und -verhalten in eine regelbasierte Form ließ sich nur in einigen Bereichen wirtschaftlich rechtfertigen, wenn es denn überhaupt möglich erschien. Regelbasierte Repräsentationen stellen enorme Anforderungen an Präzision und Vollständigkeit. Wenn das relevante Wissen nicht bis ins Detail modelliert ist, ist ein solches System kaum zu gebrauchen.

Aber auch dort, wo es gelang, mit Hunderten oder Tausenden von Regeln verblüffende Erfolge zu erzielen, gestaltete sich die Wartung der Wissensbasis als ungemein kompliziert. Viele für den Einsatz von Expertensystemen gewählte Bereiche haben eine enorme Dynamik, sei es in der Medizin oder bei der Konfigurierung von Computersystemen. Aufgrund der komplizierten Interaktion der Regeln untereinander genügt es bei Änderungen im zugrundeliegenden Bereich oft nicht, zusätzliches Wissen in Form weniger neuer Regeln hinzuzufügen – ohne Kenntnis der prozeduralen Abläufe bei der Verkettung kann dabei eine Menge Schaden angerichtet werden.

Die Kritiker

Die Unzulänglichkeiten der regelbasierten Expertensysteme waren für die Kritiker der For-

Offenbar haben die Dreyfus-Brüder aber bei ihrer Kritik das Kind mit dem Bade ausgeschüttet: Wer Regeln für Expertensysteme nicht adäquat findet, muß dabei noch lange nicht auf symbolische Informationsverarbeitung verzichten. Konkrete Erfahrungen – Fälle eben – lassen sich schließlich auch symbolisch repräsentieren. Vielleicht können sie auch helfen, menschliches Verhalten zu beschreiben. Denn was genau sich hinter der 'Intuition' verbirgt, die der Schlüssel zum Verständnis menschlicher Experten sein soll, bleibt im Dreyfus-Buch offen. Intuition sei 'das Produkt des Eingebunden-Seins in eine Situation und des Erkennens von Ähnlichkeiten' – eine Cha-

Roger Schank, schillernder Forscherstar zwischen Kognitionspsychologie und Künstlicher Intelligenz, meint sogar, daß fallbasierte Mechanismen den Kern intelligenten (und damit auch expertenhaften) Verhaltens bilden. Ende der siebziger Jahre legte er die Grundlagen für diese neue Sichtweise bei der Beschreibung und Automatisierung intelligenten Verhaltens. Dieses ist stark durch konkrete Erfahrungen beeinflusst, so Schank. Zwei Prozesse kennzeichnen danach eine typische Vorgehensweise menschlicher Experten angesichts einer Problemstellung: Sie erinnern sich an frühere ähnliche Situationen und passen die früher angewandten Lösungen an die aktuelle Situation an. An seinem damaligen Institut an der Yale University wurden die ersten

Zunächst statet man ein fallbasiertes System mit einer Sammlung typischer Fälle aus, die meist von einem menschlichen Experten bereitgestellt werden. Ihm fällt es in der Regel leichter, Beispiele seines Verhaltens zu geben, als korrek-

Fallbasierte Systeme bieten aber auch die Gelegenheit, eine neue Sichtweise auf vorhandene Daten zu gewinnen, indem be-

das System mit Wissen füllen; der Endanwender kommuniziert über eine Dialogkomponente, mit deren Hilfe er Aufgaben formuliert und ihm Lösungen übermittelt werden. Die Dialog-, Erklärungs-, Wissenserwerbs- und Inferenzkomponente sind weitgehend unabhängig vom Anwendungsbereich und werden daher zu sogenannten 'Shells' zusammengefaßt und angeboten.



Eine industrielle Anwendung

In den USA haben die ersten fallbasierten Expertensysteme den Weg in die Fabrikhallen gefunden. Bei einem großen Flugzeughersteller ist das System Clavier im Einsatz, um bei der Bestückung riesiger Brennöfen zu helfen. Im Flugzeugbau werden viele Teile aus Verbundwerkstoffen (etwa aus Graphit und Fiberglas) hergestellt, die mehrere Stunden in Öfen gehärtet werden, in denen sich Temperatur und Druck variieren lassen.

Um den teuren Ofen auszunutzen, sollen so viele Teile wie möglich auf großen Backblechen hineingeschoben werden. Dieses Problem stellt sich zur Adventszeit wohl auch in vielen deutschen Haushalten. Allerdings sind Flugzeugteile heikler als Zimtsterne, denn es vertragen sich nicht alle Teile untereinander, so daß bei einer falschen Bestückung viel Ausschuß produziert wird. Es kommt darauf an, daß sich die Teile etwa in gleichem Maße erhitzen, und dies wiederum hängt ab von Material, Größe, Gestalt und Dicke der Teile. Auch die Lage der Teile und Bleche im Ofen und die Art der benachbarten Teile und Bleche haben einen Einfluß auf das Ergebnis. Maximaler Durchsatz bei minimalen Verlusten ist also gefragt.

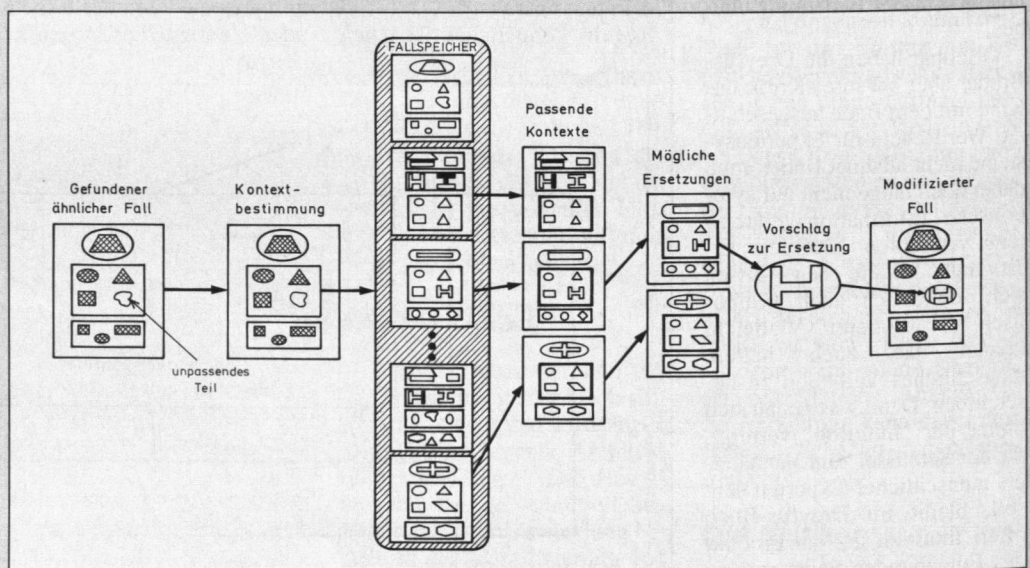
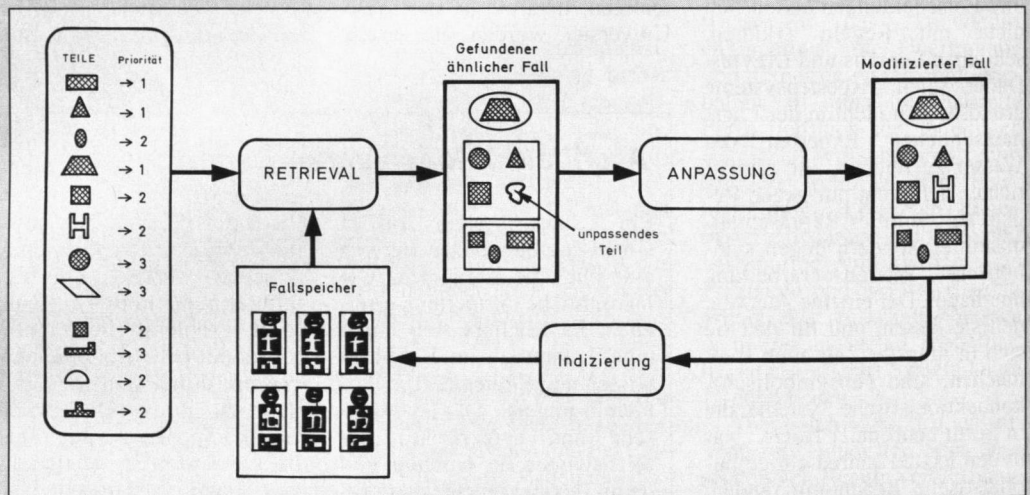
Allerdings weiß man nicht genug über das Problem, so daß eine Bestückung anhand bekannter Faustregeln mit zu großen Verlusten verbunden wäre. Bisher wurde die Aufgabe von Mitarbeitern geleistet, die anhand von Zeichnungen früherer erfolgreicher Brennvorgänge eine Ladung zusammenstellten. Da dies durch die große Zahl von Versuchen immer mühsamer wurde, lag eine Unterstützung durch ein fallbasiertes System nahe. Knowledge Engineering war bei dieser Anwendung fast überflüssig, da die Fälle schon existierten. Es entstand ein System, das tatsächlich die Schnelligkeit und Qualität der Entscheidungen mit fortwährendem Einsatz immer weiter

steigern konnte, auch im Vergleich zu dem manuellen Verfahren vorher. Schon eine Basis von zwanzig Fällen reichte aus, um Clavier am Ofen nutzbringend einzusetzen; inzwischen ist der Fallspeicher so weit gewachsen, daß die Empfehlungen eine Qualität haben, die nur noch von wenigen Experten erreicht wird. Das System wird daher auch zur Schulung von neuen Mitarbeitern eingesetzt, denen in kurzer Zeit die Erfahrungen von jahrelangen Versuchen erschlossen werden. Clavier als konstant wachsender Erfahrungsspeicher bildet einen Teil des 'corporate memory', durch

den die Firma weniger abhängig von der Fluktuation im Personalbereich ist.

Clavier arbeitet folgendermaßen: Gegeben ist eine Menge von Teilen, die im Ofen gehärtet werden sollen – typischerweise weit mehr, als in den Ofen passen. Zu jedem Teil sind Material, Größe, Gestalt und Dicke bekannt sowie ein Prioritätswert, mit dessen Hilfe Teile als besonders dringlich gekennzeichnet werden können. Gesucht wird ein Fall aus dem Fallspeicher, der besonders viele Teile mit hoher Priorität im Ofen unterbringt. Ist eine solche Konfiguration aus dem Speicher gefunden, so

muß sie meist noch an die aktuelle Situation angepaßt werden, da nur in den seltensten Fällen alle Teile der früheren Konfiguration auch im aktuellen Fall fürs Brennen vorgesehen sind. Bei der Anpassung dürfen allerdings nur solche Konfigurationen gebildet werden, von denen man aus Erfahrung weiß, daß sie wahrscheinlich erfolgreich gebrannt werden können. Der so modifizierte Fall gibt die Bestückung des Brennofens an, die von Mitarbeitern geprüft, gegebenenfalls verändert und schließlich ausgeführt wird. Die Bestückung wird nach dem Brennen hinsichtlich ihres Erfolgs bewertet,



indiziert und als neuer Fall in den Fallspeicher integriert.

Fälle anpassen

Besonderen Wert legt man bei dieser Anwendung auf Mechanismen zur Anpassung früherer Fälle, um den Ofen möglichst gut auszunutzen, auch wenn der gefundene ähnliche Fall Teile enthält, die in der aktuellen Situation überhaupt nicht gebrannt werden sollen. Diese Teile aus dem früheren Fall sollen durch solche ersetzt werden, die tatsächlich zu brennen sind. Welche der Teile, die auf die Verarbeitung im Ofen warten, sich tatsächlich sinnvoll ergänzen lassen, ist eine Frage der Verträglichkeit der Teile untereinander, was wiederum nur durch Erfahrung bestimmt werden kann. Daher verwendet Clavier an dieser Stelle einen zweiten fallbasierten Zyklus. Der erste bestand darin, wie oben dargestellt, eine Konfiguration aus dem Fallspeicher zu finden, bei der möglichst viele der auf das Brennen wartenden Teile mit hoher Priorität vertreten sind. In dem zweiten fallbasierten Zyklus sollen nun die restlichen Plätze auf den Blechen verteilt werden, die im früheren Fall mit Teilen besetzt waren, die nicht aktuell auf die Verarbeitung warten. Dabei muß das Programm den Kontext berücksichtigen, also die Art und Lage der anderen Teile sowie die Brenndauer. Als nachrangiges Kriterium wird die Priorität der Teile eingesetzt. Für die Anpassung einer Konfiguration an die aktuelle Situation werden nun bei jedem zu ersetzenden Teil drei Schritte durchgeführt: Zunächst wird der Kontext des Teils bestimmt, darauf folgt die Suche im Fallspeicher nach Fällen mit ähnlichem Kontext, und schließlich wird einer von ihnen ausgewählt und mit seiner Hilfe das Teil für den Austausch bestimmt.

stehende Datenbanken zu Fallspeichern umgewandelt werden. Fallbasierte Verarbeitungsmechanismen ermöglichen dann eine flexiblere Nutzung und Interpretation der Informationen, die über die Zugriffsmöglichkeiten herkömmlicher Datenbanksysteme hinausgehen.

Ähnlichkeiten finden

Angeichts einer neuen Aufgabe muß ein fallbasiertes Expertensystem zunächst eine ähnliche Situation aus dem Fallspeicher finden. Was 'ähnlich' heißt, hängt vom Bereich ab und muß vorher festgelegt werden. Dabei kann es sich um geringe Unterschiede bei numerischen oder symbolischen Werten handeln, die die Situation beschreiben. Oder es können aufwendigere Verfahren erforderlich sein, die zusätzliches Bereichswissen zur Identifizierung ähnlicher Situationen bereitstellen. Hier wie auch an anderen Stellen bietet es sich an, regelbasierte Mechanismen zur Unterstützung bereitzustellen. Ist nun ein ähnlicher Fall gefunden, so ist manchmal auch schon das neue Problem gelöst: Geht es beispielsweise um ein Klassifikationsproblem wie bei der Bonitätsprüfung eines potentiellen Kreditnehmers, so kann eventuell einfach die Lösung des gefundenen Falls kopiert werden. Oft ist es allerdings notwendig, die frühere Lösung zu modifizieren und der aktuellen Situation anzupassen. Schließlich wird der aktuelle Fall mitsamt seiner Lösung indiziert und in den Fallspeicher integriert und steht so als Ausgangsbasis für künftige Problemstellungen zur Verfügung. Damit sind die grundlegenden Schritte genannt, die ein System durchführen muß, um die Bezeichnung 'fallbasiert' zu verdienen.

Viele Systeme gehen aber weit über den grundlegenden Zyklus hinaus. So schließt sich an den Anpassungsschritt oft eine Erprobung der gefundenen Lösung an, wobei auftretende Fehler mit Hilfe von Bereichswissen erklärt und behoben werden. Bei erfolgreicher Reparatur kann anhand der für den Fehler verantwortlichen Merkmale der Situation der Mechanismus zur Fallindizierung verändert werden, so daß solche

Fehler in Zukunft vermieden werden.

Mit Unterstützung

Jahre nach der Erstellung der ersten regelbasierten Expertensysteme kamen die ersten 'Shells' auf den Markt, die die Implementierung solcher Systeme erleichtern, indem sie bereichsunabhängige Teile wie den Inferenzmechanismus bereitstellen und für die bereichsspezifischen Teile des Systems komfortable Möglichkeiten zur Kodierung bieten. Analog dazu wurden in den letzten Monaten die ersten fallbasierten Shells angeboten. Anhand der oben genannten Verarbeitungsschritte wird klar, in welcher Weise Shells den fallbasierten Verarbeitungszyklus unterstützen können: Zunächst einmal sollen sie Mittel zur Repräsentation von Fällen bereitstellen. Weiterhin sollen sie erlauben, die Ähnlichkeit von Fällen in vielerlei Weise zu definieren und in Verbindung damit die Indizierung des Fallspeichers durchzuführen, damit ein effizientes Retrieval gesichert ist. Schließlich sollen Mechanismen integrierbar sein, die die Anpassung von früheren Fällen an die aktuelle Situation erlauben. Viele weitere Punkte gibt es, die die Leistungsfähigkeit fallbasierter Shells erhöhen können: Etwa die Möglichkeit, im Rahmen eines hybriden Systems verschiedene Repräsentationsmittel zu kombinieren, beispielsweise regelbasierte, objektorientierte und fallbasierte Techniken. Oder die Bereitstellung von Erklärungsmechanismen, anhand derer man das Vorgehen des Systems nachvollziehen kann. Die von Roger Schank gegründete Firma 'Cognitive Systems' ist dabei ganz vorne zu finden: Ihre Shell ReMind gehört zu den ersten und leistungsfähigsten Shells auf diesem Gebiet. In einer der nächsten Ausgaben von c't werden wir solcherart Produkte vorstellen.

Die Spanne der Aufgabengebiete, in denen fallbasierte Systeme bisher erprobt wurden, ist groß: Planen, technische und medizinische Diagnose, Konstruktion und juristische Problemstellungen gehören dazu, ebenso wie der erfolgreiche Einsatz bei Lehrprogrammen. Auch in Deutschland wurden in den

letzten Jahren die ersten Systeme dieser Art entwickelt. An den Universitäten Kaiserslautern, Hamburg und Karlsruhe hat man Prototypen implementiert, die auf dem Gebiet der Konstruktion oder Fehlerdiagnose von Maschinen arbeiten. Unter dem Akronym FABEL wird seit kurzem ein Verbundvorhaben vom Bundesministerium für Forschung und Technologie gefördert, das neben einer Methodologie für die Erstellung fallbasierter Expertensysteme auch Anwendungen in den Bereichen kardiologische Diagnostik und Architektur umfaßt. Eine ganz andere Anwendung fand der Ansatz in Freiburg: Am Psychologischen Institut der dortigen Universität werden computergestützte Lehrsysteme mit der Fähigkeit ausgestattet, den Benutzer an frühere ähnliche Situationen aus dem Dialog zu erinnern. Insbesondere bei Systemen zum 'entdeckenden Lernen', bei denen der Benutzer etwa anhand von Simulationen selbst die zugrundeliegenden Gesetzmäßigkeiten herausfinden soll, kann dadurch ein effektiveres Lernen möglich werden.

Fazit

Auch wenn fallbasierte oft schneller als herkömmliche Expertensysteme zu entwickeln und flexibler einzusetzen sind, so ist im Einzelfall noch eine ganze Menge zu tun, bevor der Computer wirklich als Experte in einem Gebiet angesehen werden kann. Die Firma Inference hat mit der eingangs zitierten Werbeaussage den Mund wohl doch etwas voll genommen. Es sei denn, es handelt sich um eine Polarnacht. (ae)

Literatur

- [1] H. L. Dreyfus und S. E. Dreyfus, Künstliche Intelligenz – Von den Grenzen der Denkmachine und dem Wert der Intuition, Reinbek, Rowohlt 1987
- [2] C. K. Riesbeck und R. C. Schank, Inside case-based reasoning, Hillsdale, N J: Lawrence Erlbaum Ass. 1989. (Zum Weiterlesen sehr gut geeignet. Das Buch enthält anschauliche Einführungen in klassische fallbasierte Systeme mit vielen Programmbeispielen in Lisp.) 